AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application.

What is claimed is:

1. (Original) In a processor to handle processing of at least first and second threads in parallel, a method of assigning thread priority comprising:

assigning priority to said first thread;

loading a preliminary value to a thread precedence counter; and

assigning priority to said second thread after said thread precedence counter expires.

2. (Original) The method of claim 1 wherein said preliminary value is based on a value stored in a first starting counter associated with said first thread.

3. (Original) The method of claim 2 further comprising:

determining whether there is an indication of approaching instruction side starvation for said first thread; and

incrementing a value stored in said first starting counter is incremented when there is an indication of approaching instruction side starvation for said first thread.

4. (Currently Amended) The method of claim 3 wherein determining whether there is an indication of approaching instruction side starvation for said first thread includes

determining whether each of a plurality of conditions are true, the plurality of conditions including each of the following

whether the processor is operating in a multithreaded processing mode;

whether the first thread has no instructions in an execution pipeline of said processor; and

whether the first thread is attempting to fetch instructions from a memory.

5. (Original) The method of claim 4 wherein when incrementing the value stored in the first starting counter, said value is incremented geometrically.

6. (Original) The method of claim 5 wherein said value is incremented geometrically by left-shifting a binary 1 bit into said value.

7. (Currently Amended) In a processor to handle processing of at least first and second threads in parallel, a method of assigning thread priority comprising:

assigning priority to said first thread; and

assigning priority to said second thread when one of a pluarality of conditions is true, the conditions including:

whether processing of said first thread retires an instruction from said first thread; and

whether ~~and~~ there is not an indication of approaching instruction side starvation for said first thread.

8. (Currently Amended) The method of claim 7 wherein said indication of approaching instruction side starvation for said first thread includes each of a plurality of conditions being true, the plurality of conditions including each of the following

whether the processor is operating in a multithreaded processing mode;

whether the first thread has no instructions in an execution pipeline of said processor; and

whether the first thread is attempting to fetch instructions from a memory.

9. (Original) A processor to handle processing of at least first and second threads in parallel, comprising:

control logic to assign priority between said first and second threads;

a thread precedence counter coupled to said control logic wherein priority is assigned to said second thread after said thread precedence counter expires.

10. (Original) The processor of claim wherein said preliminary value is based on a value stored in a first starting counter associated with said first thread.

11. (Original) The processor of claim 10 wherein said control logic is to determine whether there is an indication of approaching instruction side starvation for said first thread and to increment a value stored in said first starting counter is incremented when there is an indication of approaching instruction side starvation for said first thread.

12. (Currently Amended) The processor of claim 11 wherein said control logic is determine whether there is an indication of approaching instruction side starvation for said first thread by determining whether each of a plurality of conditions are true, the plurality of conditions including each of the following

whether the processor is operating in a multithreaded processing mode;

whether the first thread has no instructions in an execution pipeline of said processor; and

whether the first thread is attempting to fetch instructions from a memory.

13. (Original) The processor of claim 12 wherein said control logic is to increment the value stored in the first starting counter geometrically.

14. (Original) The processor of claim 13 wherein said value is to be incremented geometrically by left-shifting a binary 1 bit into said value.

15. (Original) A processor to handle processing of at least first and second threads in parallel, comprising:

control logic to assign priority to said first thread and to assign priority to said second thread when one of a pluarlity of conditions is true, the conditions including:

whether processing of said first thread retires an instruction from said first thread; and

whether and there is not an indication of approaching instruction side starvation for said first thread.

16. (Currently Amended) The processor of claim 15 wherein said indication of approaching instruction side starvation for said first thread includes each of a plurality of conditions being true, the plurality of conditions including each of the following

whether the processor is operating in a multithreaded processing mode;

whether the first thread has no instructions in an execution pipeline of said processor; and

whether the first thread is attempting to fetch instructions from a memory.

17. (Original) A computer system to handle processing of at least first and second threads in parallel, comprising:

a memory to store instructions for first and second threads;

a processor including

control logic coupled to said memory to assign priority between said first and second threads;

a thread precedence counter coupled to said control logic wherein priority is assigned to said second thread after said thread precedence counter expires.

18. (Currently Amended) The computer system of claim 17 wherein ~~said~~ preliminary value is based on a value stored in a first starting counter associated with said first thread.

19. (Currently Amended) The computer system of claim 18 wherein said control logic is to determine whether there is an indication of approaching instruction side starvation

for said first thread and to increment a value stored in said first starting counter ~~is incremented~~ when there is an indication of approaching instruction side starvation for said first thread.

20. (Currently Amended) The computer system of claim 19 wherein said control logic is determine whether there is an indication of approaching instruction side starvation for said first thread by determining whether each of a plurality of conditions are true, the plurality of conditions including each of the following

whether the processor is operating in a multithreaded processing mode;

whether the first thread has no instructions in an execution pipeline of said processor; and

whether the first thread is attempting to fetch instructions from a memory.

21. (Original) The computer system of claim 20 wherein said control logic is to increment the value stored in the first starting counter geometrically.

22. (Original) The computer system of claim 21 wherein said value is to be incremented geometrically by left-shifting a binary 1 bit into said value.

23. (Original) A computer system to handle processing of at least first and second threads in parallel, comprising:

a memory to store instructions for first and second threads;

a processor including

control logic to assign priority to said first thread and to assign priority to said second thread when one of a pluarlity of conditions is true, the conditions including:

whether processing of said first thread retires an instruction from said first thread; and

whether and there is not an indication of approaching instruction side starvation for said first thread.

24. (Currently Amended) The computer system of claim 23 wherein said indication of approaching instruction side starvation for said first thread includes each of a plurality of conditions being true, the plurality of conditions including each of the following

whether the processor is operating in a multithreaded processing mode;

whether the first thread has no instructions in an execution pipeline of said processor; and

whether the first thread is attempting to fetch instructions from a memory.

25. (Original) A set of instructions residing in a storage medium, said set of instructions capable of being executed by a processor to handle processing of at least first and second threads in parallel and assign thread priority comprising:

assigning priority to said first thread;

loading a preliminary value to a thread precedence counter; and

assigning priority to said second thread after said thread precedence counter expires.

26. (Original) The set of instructions of claim 25 wherein said preliminary value is based on a value stored in a first starting counter associated with said first thread.

27. (Currently Amended) The set of instructions of claim 26 ~~wherein the method~~ further ~~includes~~comprising:

determining whether there is an indication of approaching instruction side starvation for said first thread; and

incrementing a value stored in said first starting counter is incremented when there is an indication of approaching instruction side starvation for said first thread.

28. (Currently Amended) The set of instructions of claim 27 wherein determining whether there is an indication of approaching instruction side starvation for said first thread includes determining whether each of a plurality of conditions are true, the plurality of conditions including each of the following

whether the processor is operating in a multithreaded processing mode;

whether the first thread has no instructions in an execution pipeline of said processor; and

whether the first thread is attempting to fetch instructions from a memory.

29. (Original) The set of instructions of claim 28 wherein when incrementing the value stored in the first starting counter, said value is incremented geometrically.

30. (Original) The set of instructions of claim 29 wherein said value is incremented geometrically by left-shifting a binary 1 bit into said value.